

AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application:

LISTING OF CLAIMS:

1-21. (canceled)

22. (new) A method for gene mapping chromosome and phenotype data from a database, comprising analyzing linkage disequilibrium between genetic marks  $m_i$ ,

i) searching from the data for all marker patterns  $P$  that satisfy a pattern evaluation function  $e(P)$ , wherein

a: the marker patterns are expressions within the database comprising genetic markers and their alleles and zero or more of the following: individual covariates, environmental variables and auxiliary phenotypes; and

b: the pattern evaluation  $e(P)$  is a measure of the association between the marker pattern  $P$  and a phenotype being studied,

ii) scoring each marker  $m_i$  of the data with a marker score  $s(m_i)$ , which is a function of the set  $S_i$  defined as the set of marker patterns overlapping the marker  $m_i$  and satisfying the pattern evaluation function  $e$  as defined in step i), and

iii) mapping the location of a gene by evaluating the scores  $s(m_i)$  of all the markers  $m_i$  in the data which is determined by maximizing the score if the scoring function is designed to give higher scores closer to the gene, and on minimizing the score if the scoring function is designed to give lower scores closer to the gene.

23. (new) The method of claim 22, wherein the chromosome data consists of either haplotypes or genotypes.

24. (new) The method of claim 23, wherein said haplotypes and genotypes contain flexible regions.

25. (new) The method of claim 22, wherein the marker patterns  $P$  are searched by the following algorithm:

Input

- set of  $U$  marker patterns
- evaluation function  $e(P)$  for patterns  $P$  in  $U$
- (generalization) relation  $<$  for patterns in  $U$
- where the function  $e$  and the relation  $<$  are such that if  $e(P)$  is true and  $P' < P$ , then  $e(P')$  is also true

Output

```
set S = {P ε U/ e(P) is true} of patterns
```

Method

```
S : = {}  
// Initialize the set of evaluated patterns:  
E : = {}  
// Start with the most general patterns:  
Gen := {P in U} | there is no P' in U, P' != P, such that P' < P}  
// Recursively evaluate patterns in a depth first order:  
foreach P ε Gen { evaluatePatterns(P) }  
end;  
  
procedure evaluatePatterns(P) {  
    insert P into the set E  
    if e(P) = true then {  
        insert P into set S  
        // Find all specializations of P that have not been tested yet,  
        and  
        // evaluate them recursively;  
        Spec := {P' in U-E | P < P', P' != P, and there is no P'' in U-E,  
        P''' != P  
        and P''' != P', with P < P''' < P'};  
        foreach P' in Spec {evaluatePatterns(P'); }  
    }
```

26. (new) A method of claim 22, wherein the marker patterns  $P$  are searched by the following algorithm:

Input

- set  $U$  of marker patterns
- evaluation function  $e(P)$  for patterns  $P$  in  $U$
- frequency threshold  $x$

Output

- set  $S = \{P \text{ in } U \mid e(P) \text{ and } ae(P) \text{ is true}\}$  of patterns, where  $ae(P)$  is true if and only if the frequency of pattern  $P$  exceeds a given threshold  $x$

Method

```
S : = {}

// Initialize the set of evaluated patterns:

E : = {}

// Start with the most general patterns:

Gen : = {P in U | there is no P' in U, P' != P, such that P -> P'}

//Recursively evaluate patterns in a depth first order:
```

```
foreach P in Gen {evaluatePatterns(P) }  
end  
  
procedure evaluatePatterns(P) {  
    insert P into the set E  
    if ae(P) = true then {  
        if e(P) = true then insert P into set S  
        // Find all specializations of P that have not been tested  
        yet, and evaluate  
        // then recursively:  
        Spec := {P' in U-E | P' -> P, P' != P, and there is no P'' in  
U-E, P'' != P  
        and P'' != P', with P' -> P'' and P'' -> P}  
        foreach P' in Spec {evaluatePatterns(P') }  
    }  
}
```

27. (new) The method of claim 22, wherein the marker patterns *P* are searched by the following algorithm:

Input

- marker map  $M = (m_1, \dots, m_k)$
- phenotype vector  $Y = (Y_1, \dots, Y_n)$
- haplotype matrix  $H$  of size  $n * k$   
association threshold  $x$  for chi-squared test

maximum pattern length  $l$

- maximum number of gaps  $g$
- maximum gap size  $s$

#### Output

- set  $S = \{P \text{ in } U \mid e(P) \text{ is true}\}$  of patterns,
- where  $U$  consists of patterns on  $M$  that consist of marker-allele assignments and that adhere to parameters  $l$ ,  $g$ , and  $i$ , and
- where  $e(P)$  is true if and only if chi-squared test on  $P$  using haplotype matrix  $H$  and phenotypes  $Y$  exceeds the given threshold  $x$

#### Method

```
S := {}  
// Number of case and control chromosomes:  
piA := number of disease-associated chromosomes;  
pic := number of control chromosomes;  
pi := piA + pic  
// A lower bound for pattern frequency:  
lb := piA * pi * x / (pic * pi + piA * x)  
// Variable for iterating over different patterns:  
P = (p1, ..., pk) := ('*', ..., '*')  
For i := 1 to k {
```

```
// alleles( $m_i$ ) is the set of alleles of the  $i$ :th marker

foreach  $a$  in alleles( $m_i$ ) {

 $p_i := a$ 

// Test pattern  $P$  and all its extensions:

checkPatterns( $P, i, i, O, O$ )

// Reset  $p_i$ :

 $p_i := '*'$ 

}

}

}

end

// Test haplotype pattern  $P$  and all patterns that can be generated
by extending  $P$ 

// from the right:

procedure checkPatterns( $P, start, i, nr\_of\_gaps, gap\_length$ ) {

// Output strongly associated patterns

if chi-squared( $P, M, H, Y$ )  $\geq x$  and  $p_i \neq '*'$  then insert  $P$  into
set  $S$ 

// Return if extended patterns would be too long:

if  $i = k$  or  $i+1-start > l$  then return

// Return if extended patterns cannot be strongly disease-
associated:

if frequency of  $P$  in disease-associated chromosomes is less than
 $lb$ 
```

```
then return;

// Create and test legal extensions of current pattern  $P$  (3
cases):

// 1. Give a marker  $i+1$  all possible values:

foreach  $a$  in alleles( $m_{i+1}$ ) {

 $p_{i+1} := a$ 

checkPatterns ( $P$ ,  $start$ ,  $i+1$ ,  $nr\_of\_gaps$ , 0)

}

// 2. Introduce a new gap starting at marker  $i+1$ :

if  $p_i \neq \ast$  and  $nr\_of\_gaps < g$  and  $s \geq 1$  then {

 $p_{i+1} := \ast$ 

checkPatterns ( $P$ ,  $start$ ,  $i+1$ ,  $nr\_of\_gaps+1$ , 1)

}

// 3. Extend the current gap over marker  $i+1$ :

if  $p_i = \ast$  and  $gap\_length < s$  then {

 $p_{i+1} := \ast$ 

checkPatterns ( $P$ ,  $start$ ,  $i+1$ ,  $nr\_of\_gaps$ ,  $gap\_length+1$ )

}

// Before returning, reset  $p_{i+1}$ :

 $p_{i+1} := \ast$ 

return

}
```

28. (new) The method of claim 22, wherein the marker patterns  $P$  are searched by the following algorithm:

Input

- set  $U$  of marker patterns
- evaluation function  $e(P)$  for patterns  $P$  in  $U$
- (generalization) relation  $<$  for patterns in  $U$ , where the function  $e$  and the relation  $<$  are such that if  $e(P)$  is true and  $P' < P$ , then  $e(P')$  is also true

Output

- set  $S = \{P \text{ in } U \mid e(P) \text{ is true}\}$  of patterns

Definitions

- function  $Lgg: U \rightarrow 2^U$ ,  $Lgg(P) = \{P' \text{ in } U \mid P > P' \text{ and } P' \neq P \text{ and there is no } P'' \text{ in } U \text{ such that } P \neq P'' \neq P' \text{ and } P > P'' > P'\}$ , the set of least generalizations of pattern  $P$
- function  $Lss: U \rightarrow 2^U$ ,  $Lss(P) = \{P' \text{ in } U \mid P < P' \text{ and } P' \neq P \text{ and there is no } P'' \text{ in } U \text{ such that } P \neq P'' \neq P' \text{ and } P < P'' < P'\}$ , the set of least specializations of pattern  $P$

Method

```
S : = { }

Q : = { }

// Start with the most general patterns:

F : = {P in U | there is no P' in U, P' != P, such that P' < P};

while F != {} {

    // Evaluate the candidate patterns:

    foreach P in F {

        if e(P) = true then insert P into set S

        else remove P from set F

    }

    Q : = Q union F

    // Generate a new set of candidate patterns:

    C : = {}

    foreach P in F {

        C : = C union { P' in U | P' in Lss(P) and for all

P'' in Lgg(P'):

            P'' in Q }

    }

    F := C

}

end
```

29. (new) The method of claim 22, wherein the marker patterns  $P$  are searched by the following algorithm:

Input

- set  $U$  of marker patterns
- evaluation function  $e(P)$  for patterns  $P$  in  $U$
- frequency threshold  $x$

Output

- set  $S = \{P \text{ in } U \mid e(P) \text{ and } ae(P) \text{ is true}\}$  of patterns, where  $ae(P)$  is true if and only if the frequency of pattern  $P$  exceeds a given threshold  $x$

Definitions

- function  $Lgg: U \rightarrow 2^U$ ,  $Lgg(P) = \{P' \text{ in } U \mid P \rightarrow P' \text{ and } P' \neq P \text{ and there is no } P'' \text{ in } U \text{ such that } P \neq P'' \neq P' \text{ and } P \rightarrow P'' \rightarrow P'\}$ , the set of least generalizations of pattern  $P$
- function  $Lss: U \rightarrow 2^U$ ,  $Lss(P) = \{P' \text{ in } U \mid P' \rightarrow P \text{ and } P' \neq P \text{ and there is no } P'' \text{ in } U \text{ such that } P \neq P'' \neq P' \text{ and } P' \rightarrow P'' \rightarrow P\}$ , the set of least special specializations of pattern  $P$

Method

```
S : = {}

Q : = {}

// Start with the most general patterns:

F := {P in U | there is no P' in U, P' != P, such that P -> P' };

while F != {} {

    // Evaluate the candidate patterns:

    foreach P in F {

        if ae(P) = true then

            if e(P) = true then insert P into set S

        }

        else remove P from set F

    }

    Q : = Q union F

    // Generate a new set of candidate patterns:

    C : = {}

    foreach P in F {

        C : = C union { P' in U | P' in Lss(P) and for all

P'' in Lgg(P'):

            P'' in Q }

    }

    F : = C

}
```

end

30. (new) The method of claim 22, wherein

- a) the phenotype being studied is qualitative, and
- b) the pattern evaluation function  $e(P)$  has the form  $e(P) = \text{true if and only if } e'(P) > x$ , where  $e'(P)$  is the (signed) association measure  $\chi^2$  and  $x$  is a user specified minimum value, and
- c) the score  $s(m_i)$  of marker  $m_i$  is the size of  $S_i$ , also called marker-wise pattern frequency of  $m_i$  and denoted by  $f(m_i)$ .

31. (new) The method of claim 22, wherein

- a) the pattern evaluation function  $e(P)$  has the form  $e(P) = \text{true if and only if } e'(P) > x$ , where  $e'(P)$  is the absolute frequency of pattern  $P$  in the data and  $x$  is a user-specified value,
- b) in order to derive the score  $s(m_i)$ , the p value (statistical significance) of each marker pattern  $P$  in determining the phenotype being studied is evaluated, and
- c) the score  $s(m_i)$  is the distance between the

observed p value distribution of patterns in  $S_i$  and the uniform distribution, defined as average of  $(p_i - q_i) \log (p_i / q_i)$  over all  $i = 1..n$ , where  $n$  is the number of haplotype patterns in  $S_i$ ,  $p_i$  is the  $i$ th smallest p value in  $S_i$ , and  $q_i$  is the expectation of the  $i$ th smallest p value, if the p values were randomly drawn from the uniform distribution.

32. (new) The method of claim 31, where the p value is computed using a linear model of form  $Y = \beta_1 X_1 + \dots + \beta_k X_k + \alpha Z + \beta_0$ , where the dependent variable  $Y$  is the phenotype being studied,  $X_1$  through  $X_k$  are covariates, and  $Z$  is a dummy variable for the occurrence of the haplotype pattern, and the coefficients  $\alpha$  and  $\beta_*$  are adjusted for best fit, and then the significance of  $Z$  as a covariate is assessed using a t test with the null hypothesis " $\alpha = 0$ ".

33. (new) The method of claim 22, further refining each score  $s(m_i)$  by replacing it by the marker-wise p value of the score  $s(m_i)$ , where the statistical significance of  $s(m_i)$  is measured against the null hypotheses that there is no gene effect.

34. (new) The method of claim 22, wherein an area returned from a prediction of a gene location is contiguous or

fragmented or a point.

35. (new) The method of claim 22, wherein the location of the gene, predicted as a function of the scores  $s(m_i)$  and based on maximizing or minimizing the score, is predicted by the combination of most probable intervals for containing a trait-susceptibility locus that covers a proportion  $t$  ( $t \in \{0, 100\%\}$ ) of the original region obtained by taking all such points in the studied chromosomal region whose nearest marker is within the  $k$  best scoring markers, and wherein  $k$  is selected so that the resulting area has length at most  $t$  times the length of the studied region.

36. (new) The method of claim 22, wherein the location of the gene, predicted as a function of the scores  $s(m_i)$  and based on maximizing or minimizing the score, is predicted to those points in the studied chromosomal region whose nearest marker scores at least  $y$  or at most  $y$ , where  $y$  is scoring function dependent and is selected so that the probability of the gene being close to the marker is determined.

37. (new) The method of claim 22, wherein the location of the gene, predicted as a function of the scores  $s(m_i)$  and based on maximizing or minimizing the score, is determined by

evaluating the marker scores or by visualization.

38. (new) The method of claim 22, further comprising searching for multiple genes by evaluating marker patterns that refer to several potential gene loci at the same time.

39. (new) A computer-readable data storage medium having computer-executable program code stored thereon operative to perform the method of claim 22 when executed on a computer.

40. (new) A computer system having executable program code that performs the method of claim 22.